ＭＰＣ－８１６に関するＦＡＱ その２

Last Update 99/03/10

■ＩＮＤＥＸ

07600	MPC-816K	ＭＰＣ－８１６Ｋ資料(980508)
07700	MPC-816K	ＭＰＣ－８１６Ｋの出荷について(980508)
07800	MPC-816	コネクタ(980508)
07900	MPC-816	ビットチェックサンプルプログラム(980508)
08000	MPC-816K	改版 3.50g K8b1(980609)
08200	MPC-816	ＭＰＣとＭＰＧの直結方法(980616)
08300	MPC-816K	No State ｴﾗｰ、RUN 後ﾘｾｯﾄ状態、動作不安定(980626)
08400	MPC-816X(Z 版)	No State ｴﾗｰ/特定の動作を繰り返すと PRG 停止、全ての出力ＯＮ(980701)
08500	MPC-816K(Z 版)	プログラムが実行できない(980701)
08600	MPC-816K(P 版)	改版 KEY,PLS_MIF,YPLS,ZPLS,WPLS,VPLS 追加（REV-3.50j(K8b4)） (980805)
08700	MPC-816	デジタル 「ＧＰ７０シリーズ」のメモリリンク通信 サンプル (980824)
08800	MPC-816	ＭＰＣ＋ＭＰＧ＋ＭＣＴでカウンタが正常でない (980902)
08900	MPC-816K	816X で正常に動作していた MIO-240E の出力が 816K で挙動不審(980904)
09000	MPC-816K	GET#(0)で１ﾊﾞｲﾄ以上の数値を入力する（GETN#の代用）(980908)
09100	MPC-816K	改版 GETN# BUGFIX(980910)
09200	MPC-816K	プログラム実行時間(981021)
09300	MPC-816	ポイントデータの演算とコピー(981028)
09400	MPC-816	Ａ／Ｄ、Ｄ／Ａコンバーターについて(981216)
09500	MPC-816	ＹＰＬＳのスローアップ、スローダウンについて(981224)
09600	MPC-816K	ＭＰＣ－８１６Ｋのプログラム実行スピード(990118)
09700	MPC-816K	アスキーコード文字列を２バイト１０進数に変換(990310)

■■ ＭＰＣ－８１６Ｋ資料 ■■

１）概要

ＭＰＣ－８１６ＫはＭＰＣ－８１６Ｘの機能を継承した、高速ＣＰＵボードです。８１６Ｘと差し替えて使用することができますが、以下の相違点に注意して下さい。

☆実行速度

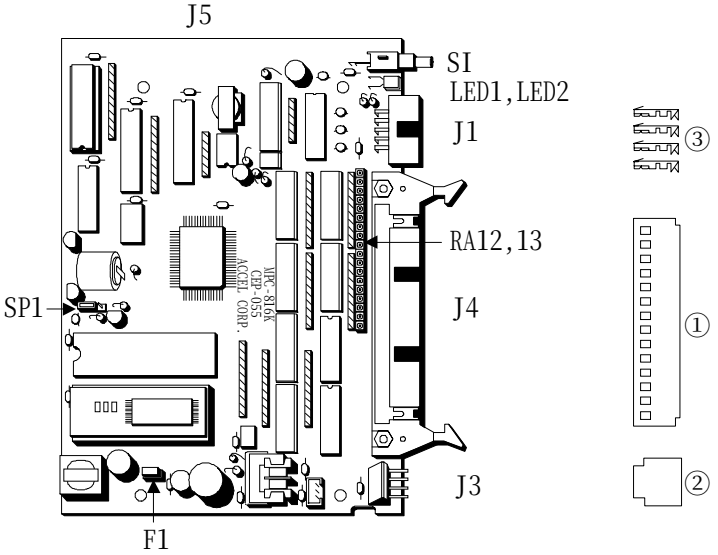
ＭＰＣ－８１６ＸのＣＰＵはＫＬ５Ｃ８０Ａ１６を用いており、旧８１６Ｘに対して約三倍の実行速度となります。これはＫＬ５Ｃ８０ＡがＺ８０互換でありながら効率の良いエンジンを備えており、１インストラクション、１クロックで実行されるためです。

☆Ｉ／Ｏアドレス（マシン語）

入出力回路（制御ＩＯ，電源、ＲＳ－２３２Ｃ）のコマンド（ＯＮ，ＯＦＦ，ＳＷ，ＩＮ，ＯＵＴ等）からの制御は８１６Ｘと完全に互換です。しかし、マシン語レベルでは入出力回路のアドレスが変更されています。このためマシン語モジュールを組み込んで使用される場合には以下のアドレスにしたがって下さい。

ＩＮ ０－７	３ＡＨ
ＩＮ ８－１１	３ＣＨ（下位４ビット）
ＩＮ １２－１５	３８Ｈ（上位４ビット）
ＯＵＴ ０－３	３８Ｈ（下位４ビット）
ＯＵＴ ４－７	３ＣＨ（上位４ビット）

２）ＭＰＣ－８１６Ｋ各部の名称と役割



添付品

① ＰＳ－Ｄ４Ｃ５０	JAE	×１
② 電源コネクタ(H4P-SHF-AA)		×１
③ コンタクト(BHF-001T-0.8SS)		×４

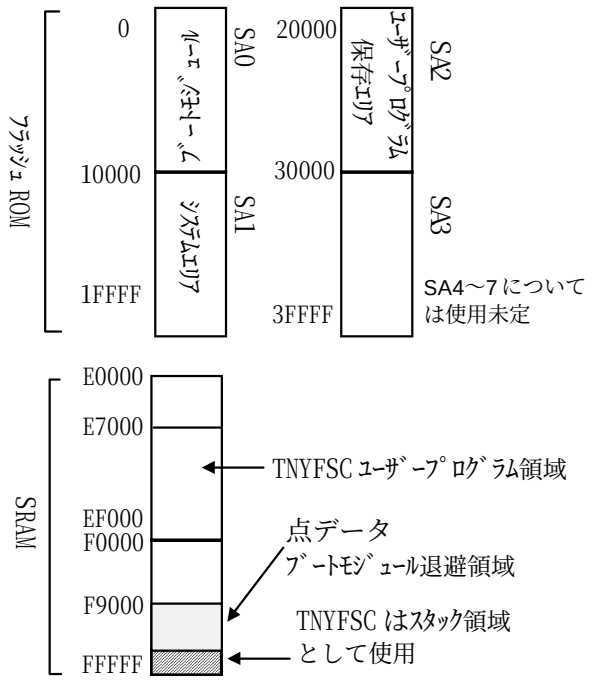
J1:ＲＳ－２３２Ｃコネクタ	プログラミング用とユーザ用があります。フォトカプラーで内部５Ｖ系と絶縁されています。
J3:電源コネクタ	ＤＣ１２Ｖ～ＤＣ２４Ｖを供給します。
J4:Ｉ／Ｏコネクタ	入力１６点、出力８点です。（入力ポート０～１５／出力ポート０～７）
J5:バスコネクタ	ＭＩＦ－８１６等と接続されます。
S1:リセットスイッチ	ＭＰＣをリセットします。
LED1,2:ＬＥＤ　青	電源が供給されていると青いＬＥＤ点灯
〃　赤	エラーが発生すると赤いＬＥＤ点灯チェックサムエラーなど
F1:ヒューズ	ＤＣ５Ｖ電源ヒューズです。（１Ａ）　ＦＫＳ５－１Ａ（浜井）
SP1:ショートピン	ＳＲＡＭ保持用のショートピンです。通常ショートです。
RA12,13:ＳＩＰソケット	入力ポートを二線式センサに対応させるための抵抗アレイ用

３）ハード構成

ＣＰＵ	ＫＬ５Ｃ８０Ａ１６ＣＦ（７．３７２５ＭＨＺ）	
メモリ	ＳＲＡＭ	ＨＭ６２８１２８ＬＰ－７　相当（バッテリバックアップ　７年以上）
	ＦＲＯＭ	２９Ｆ０４０Ａ－９０　相当
ＲＳ－２３２Ｃ	ＣＨ０	プログラム用　９６００ｂｐｓ７ｂｉｔ＋Ｅｖｅｎパリティ固定
	ＣＨ１	ユーザ用　ＣＮＦＧ＃コマンドによって設定
出力ポート	８点	オープンコレクタ　各点２００ｍＡ
入力ポート	１６点	フォトカプラ入力　各点２ｍＡ～４ｍＡ
電源	ＤＣ１２Ｖ～２４Ｖ単一	消費電流（１００ｍＡ～２００ｍＡ）

４）ブロックダイアグラムとフラッシュＲＯＭシステムの方法について

メモリマップ



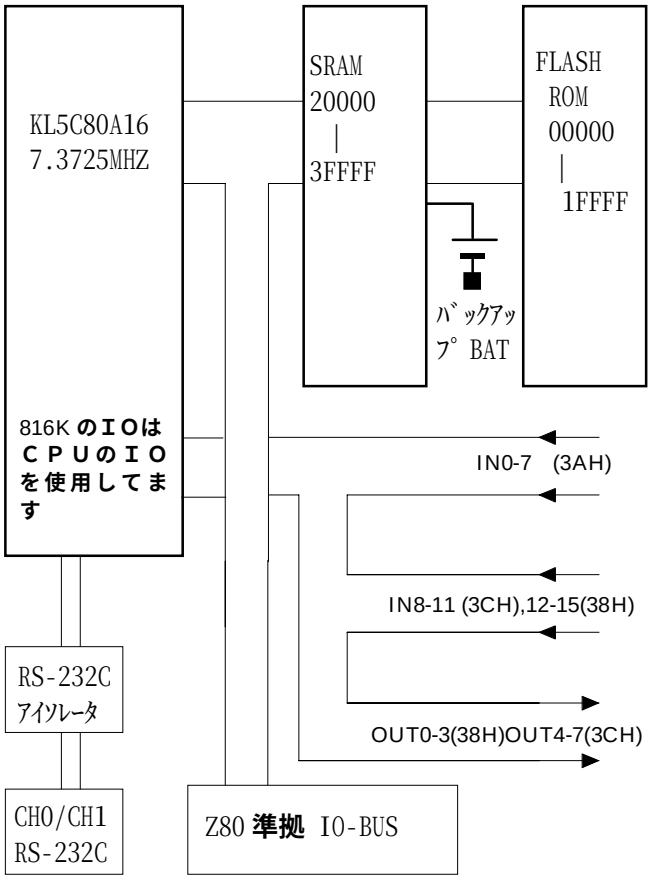
☆ブートモジュールについて

ＭＰＣ－８１６Ｋはパワーオン後ブートモジュールを起動します。ブートモジュールは書き換え管理プログラムをＳＲＡＭのブートモジュール領域に複写して制御をこちらに引き渡します。この後で、システムエリアに正しいシステムが登録されているかテストし正しければ、こちらに制御を引き渡します。この時、システムエリア(TNYFSC)は４００Ｈ－６ＦＦＦＨに配置されます。

☆システムの書き換え方法

システムエリアが正しく書き込まれていなければ、モニターがそのまま常駐しＳＡ１のイレース、システムのダウンロードを行うことができます。システムが正しく書き込まれていて、なお且つシステムを変更する場合は、電源投入後ブートモジュールが０３コードを出力しますので、この時０３を０．５秒以内にエコーバックすると、システムに制御を引き渡さず、モニターが動作しシステムの入れ替えができるようになります。これらの一連の手続きはシステムローダによって行うことができます。これによりＲＯＭ交換を必用としないメンテナンスを行うことができます。

内部ブロック図



Ｏは２４点用意されていますが、うち３８Ｈの下位４ビットが出力専用、上位４ビットが入力専用に割り当てられているために、割付が変則的になっています。このため、マシン語モジュールでのポートの制御には注意が必用となります。

ＳＲＡＭはＳＰ１をショートするとバッテリバックアップされます。

Ｊ４のＩ／Ｏコネクタのインターフェース、取扱い使用方法は５）でＭＩＦ－８１６、ＭＩＯ－８１６と併せて解説します。

ＭＰＣ－８１６Ｋには、スイッチングタイプの３端子電源が備えられておりＪ３から供給されたＤＣ２４Ｖ（ＤＣ１２Ｖ）を５Ｖに変換しＥＭＩフィルタを経て内部ＣＰＵ回路に供給しています。このＤＣ５ＶがＪ５を介してＭＩＦ、ＭＩＯに供給されています。電源の能力は最大で３８０ｍＡです。ＣＰＵボードに実装されたヒューズはこの電源を保護するためのものです。ＤＣ２４Ｖは、Ｊ４のＩ／Ｏ入出力の駆動にも使用されています。ＳＲＡＭはリティウム電池により電源オフ時にもデータが保持されています。

ＭＰＣ－８１６Ｋはきわめて簡単な構成です。ＩＯはＣＰＵ上のＩＯが割り当てられています。ＫＬ５Ｃ８０ＡのＩ

5）ＭＰＣ－８１６Ｋピンアサイン表

J1

▶	1	SG	2	TX0
	3	RX0	4	SG
	5	MAN	6	P5
	7	SG	8	TX1
	9	RX1	10	FG

※プログラム時５と６をショート、Ｐ５はＤＣ５Ｖが１ＫΩを介して出力
フォトカプラーで内部５Ｖ系と絶縁されています。

J3

1	+DC
2	
3	GND
4	FG

※＋１２≦ＤＣ≦２４　±１０％

J4

	A		B
▶ 1	入力 0	1	GND
2	1	2	GND
3	2	3	GND
4	3	4	GND
5	4	5	GND
6	5	6	GND
7	6	7	GND
8	7	8	GND
9	8	9	GND
10	9	10	GND
11	10	11	GND
12	11	12	GND
13	12	13	GND
14	13	14	GND
15	14	15	GND
16	15	16	GND
17	+DC	17	出力 0
18	+DC	18	1
19	+DC	19	2
20	+DC	20	3
21	+DC	21	4
22	+DC	22	5
23	+DC	23	6
24	+DC	24	7
25		25	SG

［バスアサイン］

J5

1	P5	2	P5
3	SG	4	SG
5	CLK	6	
7	(ME)	8	(IOE)
9	(RD)	10	(WR)
11	NC	12	SG
13	NC	14	(RST)
15	SG	16	D0
17	D1	18	D2
19	D3	20	D4
21	D5	22	D6
23	D7	24	SG
25	A0	26	A1
27	A2	28	A3
29	A4	30	A5
31	A6	32	A7
33	NC	34	NC
35	NC	36	NC
37	NC	38	NC
39	NC	40	NC

※ＣＬＫ7．3725ＭＨＺ
※ＳＧはコンピュータグラウンドです。
ＴＴＬインターフェース以外は
使用しないで下さい。
※（）付信号は、“Ｌ”アクティブを示す

ＭＰＣ－８１６Ｋの出荷について

９７年より文書によりお知らせしてきたましたが、今回納入分よりＭＰＣ－８１６Ｘに代わってＭＰＣ－８１６Ｋにて納品させて頂きます。
ＭＰＣ－８１６Ｘは使用ＩＣの製造中止にともない、生産を続けることが困難になった為です。８１６Ｋは８１６Ｘに対して電気的特性、コネクタ、プログラムの実行等について互換性を確保して設計・製作されていますので、８１６Ｘと同様にご使用頂けると思います。相違点、改良された点は以下の通りです。

☆実行速度が８１６Ｘに比較して３倍と高速です。
☆Ｚ，Ｐのバージョン切り替えが**Ｖ＿ＳＷＡＰ**コマンドにより簡単に行うことができます。尚、**Ｖ＿ＳＷＡＰ**実行後、１度だけ、パワーオンリセット、**ＭＰＣＩＮＩＴ，ＥＲＡＳＥ**を実行してください。
☆尚、Ｐ版では**ＡＲ＜ｒｅｔ＞，Ｍ　［n］　＜ｒｅｔ＞**コマンドが追加され、配列変数の状態が一覧しやすくなりました。

一部のお客様に８１６Ｋの互換性の評価を頂きました。改めて御礼申し上げます。また、実際にご使用された場合で、万一不具合などありましたら弊社技術係りまでお知らせ下さい。

コネクタ(980508)

【ＪＡＥ バラ線】

５０ピンハウジング	ＰＳ－Ｄ４Ｃ５０
２６ピンハウジング	ＰＳ－Ｄ４Ｃ２６
１６ピンハウジング	ＰＳ－Ｄ４Ｃ１６
１０ピンハウジング	ＰＳ－Ｄ４Ｃ１０
コンタクト	０３０－５１３０４－００１　（ＡＷＧ＃２４～＃２８）
手動式圧着工具	ＣＴ１５０－１－ＰＳＳＦ　（ＡＷＧ＃２４～＃２８）

【ＪＡＥ 圧接】

５０ピン	ＰＳ－５０ＳＥＮ－Ｄ４Ｐ１－１Ｃ （クローズドエンドタイプ）

【ヒロセ バラ線】

５０ピンハウジング	ＨＩＦ３ＢＡ－５０Ｄ－２．５４Ｃ
コンタクト	ＨＩＦ３－２２２６ＳＣＡ （ＡＷＧ＃２２～＃２６）

【ヒロセ 圧接】

５０ピン	ＨＩＦ３ＢＡ－５０Ｄ－２．５４Ｒ

07900　MPC-816

ビットチェックサンプルプログラム(980508)

被検査変数はＶです。
１バイト中のどれか１つのビットが１であるという前提で検査します。
結果はＩ２に入ります。
（こんな方法しか思いつかなかった（@_@）ｳﾆｳﾆ）

```
10      *LOOP
20      INPUT V
30       I1=1
40       I2=0
50      FOR I=1 TO 8
60       IF V<>I1 THEN *PASS
70        I2=I
80       *PASS
90        I1=I1*2
100     NEXT I
110     PRINT I2
120     GOTO *LOOP
```

```
>RUN
?0
 0
?1       ←V
 1       ←ビット
?2
 2
?4
 3
?8
 4
?16
 5
?32
 6
?64
 7
?128
 8
?
```

08000　MPC-816K

改版 3.50g K8b1(980609)

TNYFSC(R) Rev-3.50g [VER-PmaX2044]
 Copyright(C)by ACCEL CORP/BC-SOFT
 [300p MPC-816K MPG MODE5|6]K8b1

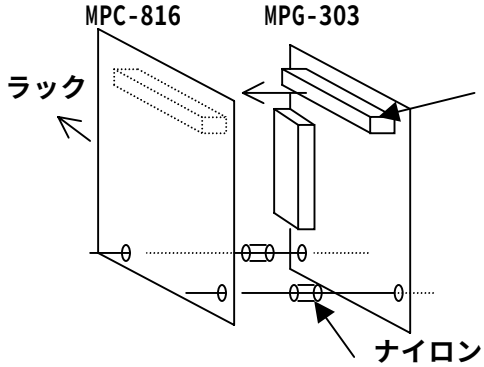
3.50g　YPLS のバグフィクス。マルチタスク下でＩＯの矛盾が出るのをバグフィクス
K8b1　　マルチタスクのスイッチのバグフィクス。期待通りの応答が出なかったものを、修復

08200　MPC-816

ＭＰＣとＭＰＧの直結方法(980616)

ＭＰＣとＭＰＧを結合するバスコネクタ（ＭＩＦのＪ２に相当）は標準実装されていませんので、直結する場合はコネクタをハンダ付けして下さい。

①ＭＰＧ－３０３にバスコネクタをハンダ付けします。
②スペーサーを入れてＭＰＣ－８１６とＭＰＧ－３０３を合わせてラックに入れます。



Ｊ５ 後付けバスコネクター １個
ヒロセ Ａ１－４０ＰＡ－２．５４ＤＳＡ 相当

ナイロンスペーサー ２個
品川商工（株）ＭＳＰＬＳ－７
（穴とﾊﾟﾀｰﾝが近いので必ず樹脂製のｽﾍﾟｰｻｰを使って下さい）

08300　MPC-816K

No State ｴﾗｰ、RUN 後ﾘｾｯﾄ状態、動作不安定(980626)

【不具合内容】

ＲＥＶ２．５０ｂ Ｋ８ａ９、ＭＯＤＥ ２でＮｏ Ｓｔａｔｅエラーなどの不具合。最悪プログラム破壊。

【原因／対策】

原因はＭＯＶＥコマンドの第３パラメータの省略でした。

ＭＯＶＥ Ｘ、Ｙ（２） を ＭＯＶＥ Ｘ、Ｙ（２）、０ というようにするとエラーは発生しません。

ＲＥＶ２．５０ｂ Ｋ８ｂ１ではエラーになりませんが、ＭＯＶＥやＲＭＯＶの第３パラメータは省略しないで下さい。

【お詫びと訂正】

マニュアルには省略可の記述がありますが、これは全て変更して省略不可として下さい。

ＭＯＶＥ、ＲＭＯＶ以外のコマンドも同様です。

次回改訂時には直します。

08400　MPC-816X

No State ｴﾗｰ/特定の動作を繰り返すと PRG 停止、全ての出力ＯＮ(980701)

【機種】
ＭＰＣ－８１６Ｘ　ＬＮ５１９　Ｚ版で使用

【不具合内容】
実行後、数回ループを繰り返すとエラー停止して、全ての出力がＯＮになる。

```
NO State!!
>MON
        TASK 0 # 21896   TASK 1 # 2111        ←TASK 0 は存在しない文番号
        TASK 2 # 3010    TASK 3 # 4300
        TASK 4 # 5010    TASK 5 # 6035
        TASK 6 # 7010    TASK 7 # 8010
```

【原因】
ＦＲＯＭ、ＲＥＶによる変化なし。他のＭＰＣ－８１６Ｘや８１６Ｋでは再現しない。
不明。おそらくメモリ不良。

08500　MPC-816K(Z 版)
プログラムが実行できない(980701)

【機種】
ＭＰＣ－８１６Ｋ Ｚ版

【不具合内容】
ＲＵＮするとプログラムが停止して、ＶＥＲデータが表示される。ＢＡＴ Ｅｒｒｏｒが表示されるときもある。

```
RUN
TNYFSC(R) Rev-2.50b [VER-Zmx1729]
 Copyright(C)by ACCEL CORP/BC-SOFT
 [255point MPC-816K MODE1~4]K8b1
!! BAT Error
> (ctrl+M)
TASK1 80        MODE 2   ←表示内容はウソ
TASK2 80        MODE 2
TASK3 80        MODE 2
TASK4 80        MODE 2
TASK5 80        MODE 2
TASK6 80        MODE 2
TASK7 80        MODE 2
```

【原因】
ＦＯＲＫの後にＡＣＣＥＬを実行していたことによる。

```
80      MODE 2
90      SETVAR A0,Z,0
110     FORK 1,2000
120     FORK 2,3000
130     FORK 3,4000
140     FORK 4,5000
150     FORK 5,6000
160     FORK 6,7000
170     FORK 7,8000
180     ACCEL 6000,300,200
190     FEED 2
```

【対策】
ＡＣＣＥＬをＦＯＲＫの前に実行する。

```
80      MODE 2
85      ACCEL 6000,300,200
90      SETVAR A0,Z,0
```

もともとＺ版のＡＣＣＥＬはマルチタスクでの使用を考えていない（プログラムの先頭に置くことを前提にしている）。ＡＣＣＥＬ実行中のメモリ操作とマルチタスクのタイマー割りこみが重なり障害。
２．５０ｄ ８Ｋｂ１ で対策（７／１現在β扱い）

改版 KEY,PLS_MIF,YPLS,ZPLS,WPLS,VPLS ｺﾏﾝﾄﾞ追加 (REV-3.50j(K8b4)) (980805)

■書式

KEY n

n:ロック番号

■解説

キィーコマンドは、フラッシュＲＯＭにロックをかけて変更を阻止するコマンドです。

KEY 9812

とコマンドを実行すると 9812 をパスワードとしてロックがかかり、フラッシュＲＯＭ上のデータは変更することができなくなります。パスワートは 2byte の整数の範囲で設定してください。解除は同様に

KEY 9812

とします。番号を忘れてしまった場合は解除できなくなります。解除方法については各ユーザで一定の取り決めをしていただきシステム管理者として登録された方以外には公開できませんのでご了承ください。なお、キィーコードに負の数を設定するとＬＩＳＴ表示もしなくなります。

******

■書式

PLS_MIF

■解説

ZPLS,YPLS,WPLS,VPLS の出力先は通常"out 0"ですが、このコマンドによって MIF パルスポートに変更することができます。この設定はパワーオンリセット、もしくは PLS_MIF -1 で解除できます。

******

■書式

YPLS p,r,c
ZPLS p,r,c

p:出力ﾎﾟｰﾄ (out0 か MIF-816 のパルスポート)
　CW =1,4,16,64　(1=X,4=Y,16=U,64=Z)
　CCW=2,8,32,128 (2=X,8=Y,32=U,128=Z)
r:ﾊﾟﾙｽﾚｰﾄ　10pps-1500pps 程度
c:ﾊﾟﾙｽｶｳﾝﾀｰ　省略可。いくつパルスを出したかを記録する変数

■解説

引数はかならず、変数で指定します。コマンド実行前に第一、第二引数の値を決めておかないと正しく動作しません。第一引数、第二引数を動作中に変更するとパルス出力を停止、あるいはポート変更できます。また、第二引数に指定した変数を変更することによって、パルスレートも動的に変更できます。
ZPLS は YPLS と同等のコマンドで YPLS,ZPLS は同時に使用することができます。停止待ちはクリアしたポート変数が２５６になるのを待ちます。パルスはデューティー５０％の正方波形です。

```
>LIST                                          >RUN
10 A=1                                          100
20 B=100                                        200
25 C0=0                                         300
30 YPLS A,B,C0                                  400
40 FOR I=1 TO 10                                500
42 I0=100*I                                     600
45 PRINT I0                                     700
50 WAIT C0=I0                                   800
55 A=0              ←停止                       900
60 WAIT A=256       ←停止待ち                   1000
65 A=1                                         >
70 YPLS A,B,C0
80 NEXT I
90 A=0
```

******

■書式

WPLS p r c
VPLS p r c

p:出力ﾎﾟｰﾄ (out0 か MIF-816 のパルスポート)
CW =1,4,16,64 (1=X,4=Y,16=U,64=Z)
CCW=2,8,32,128 (2=X,8=Y,32=U,128=Z)
r: オフパルス幅 2.17μ秒*r (次の変数がオンパルス幅 例えば r が B0 なら B1 がｵﾝﾊﾟﾙｽ)
c:ﾊﾟﾙｽｶｳﾝﾀｰ 省略可。いくつパルスを出したかを記録する変数

■解説

WPLS,VPLS はＰＷＭライクなコマンドです。周波数設定方法のほかは、YPLS と同様の使用方法になります。注意としてＹＰＬＳ,ＷＰＬＳはタイマー３，ＺＰＬＳ、ＶＰＬＳはタイマー２を共有しているため、それぞれ排他的にしか使用できません。

WPLS A,B0,C0

A はポート指定です。ＹＰＬＳと同様です。
B0 はオフ時間を規定します。時間の単位は 2.17μ秒です。50-65535(0.1m 秒-0.14 秒)まで指定できます。オン時間を規定するのは次のアドレスの変数です。この場合 B1 がオン時間指定変数です。
50 以下を指定するとパルスが正常でなくなったり、他のコマンドを受け付けなくなります。
C0 はカウンタの指定です。停止方法は停止待ちの方法は YPLS と同様です。

```
'************
 'WPLS TEST
 'SAVE WPLS01
 ' 980805
 '************
 PLS_MIF
 FORK 1,*TASK1
 '===========
*LOOP0
 'direction
 A0=1                ←方向 X-CW
 'counter
 C0=0                ←ｶｳﾝﾀｰｸﾘｱ
 '
 FOR I=1 TO 10
 'off pulse
 B0=I*100            ←ｵﾌﾊﾟﾙｽ幅
 'on pulse
 B1=I*100            ←ｵﾝﾊﾟﾙｽ幅
 WPLS A0,B0,C0
 I1=I*1000
 WAIT C0=I1
 NEXT I
 A0=0                ←停止
 WAIT A0=256         ←停止待ち
 TIME 100
 GOTO *LOOP0
```

```
 '============
*TASK1
 'direction
 A5=4                ←方向 Y-CW
 'off pulse
 B5=200              ←ｵﾌﾊﾟﾙｽ幅
 'on pulse
 B6=200              ←ｵﾝﾊﾟﾙｽ幅
 'counter
 C5=0                ←ｶｳﾝﾀｰｸﾘｱ
 VPLS A5,B5,C5
 WAIT C5=1000
 A5=0                ←停止
 WAIT A5=256         ←停止待ち
 TIME 50
 '
 A5=8                ←方向 Y-CCW
 C5=0                ←ｶｳﾝﾀｰｸﾘｱ
 VPLS A5,B5,C5
 WAIT C5=-1000
 A5=0                ←停止
 WAIT A5=256         ←停止待ち
 TIME 50
 GOTO *TASK1
```

******

■注意

ＹＰＬＳ、ＺＰＬＳ、ＷＰＬＳ、ＶＰＬＳをＭＰＣ－８１６Ｋの出力ポートで使用する場合は、ドライバとの接続に注意して下さい。ＭＰＣと共通電源（ＤＣ２４Ｖ）でドライバを駆動する場合、２Ｋ程度の抵抗を入れて駆動電流を１０ｍＡ程度に調整してください（一般的なフォトカプラ入力の場合）。ＴＴＬ／差動入力の場合はフォトカプラ等でＩ／Ｆして下さい。
またＤＣ２４Ｖ駆動の場合､波形のなまりによる位置ズレが起こる場合があります｡パルスモータ制御は PLS_MIF でＭＩＦパルスポートに割り当てて下さい。

08700　MPC-816
デジタル 「ＧＰ７０シリーズ」のメモリリンク通信 サンプル (980824)

ＲＳ－２３２ＣユーザーＣＨとデジタル社「プログラマブル表示器 ＧＰ７０シリーズ」のメモリリンク通信のサンプルです。
メモリリンクはＭＰＣがホストとなり、ＧＰメモリの読込み／書込みでパネルをコントロールします。

●ＳＡＭＰＬＥ １を実行するには、ＧＰ側を「割り込み出力」するようにデザインします。
「割り込み出力」とは「ワードＳＷ」や「Ｔタグ」のワードアドレスをシステムデータエリア「１３」に割り当てると、それを押したときに「定数」をＲＳ－２３２Ｃ出力するというものです。
各ワードＳＷやＴタグに固有の「定数」を設定しておけば、押されたものの識別が出来ます。
ＭＰＣはその出力をＧＥＴ＃で待っているだけです。
ランプ等のＭＰＣからの出力はＧＰの「W」コマンドで行っています。
この方法の利点は、プログラムが簡単・常時ループしない・ＧＰのボタンを押したときのレスポンスが良い、などがあります。
しかし、出力可能な定数は１バイト（＆ＨＦＦは不可）なので、これを超えるデータは得られません。
ＭＰＣとリンクできる部品の種類と数も限られます。簡単なパネルデザイン向きです。

●ＳＡＭＰＬＥ ２はパネルからの読込みをＧＰの「R」コマンドで行っています。
書込みはＳＡＭＰＬＥ １と同じＧＰの「W」コマンドです。
この方法の利点はワードデータが扱えることです。
ＧＰの「数値表示器」や「設定値表示器」などの１６ビットデータの読込みができます。
欠点は、プログラムが複雑、常にループする、一度に多くのデータを入力するとレスポンスが低下するです。
レスポンスについては、ページ単位で入力範囲を制限すれば良いでしょう（同一ページの部品のアドレスは１回のＲＥＡＤで読める範囲に設定する）。このサンプルでは１ループで４ワード（６４ビット）をＲＥＡＤしていますが、この程度ならレスポンスの低下はあまり感じられません。

●ＧＰの通信設定
通信プロトコル： アスキー互換
通信設定： 伝送速度９６００ｂｐｓ、データ長８ｂｉｔ、ストップビット１、パリティ無
通信方式： ＲＳ－２３２Ｃ

※メモリリンクプロトコル、ＧＰ操作、ＧＰ作画についてはデジタル社の各取扱説明書をご覧下さい。

●ＳＡＭＰＬＥ １

```
 '************
 ' GP-H70
 '  SAMPLE
 '------------
 ' USE T-TAG
 ' & SYS Adr13
 '============
 'GPH70S10
 '  980818
 '************
 CNFG# 4,0,2
 FORK 1,*RS_RCV  "入力用のタスク
*LP1
 A=600           "書込みアドレス
 S=IN(0)"書込みデータ
 GOSUB *WRITE
 WAIT IN(0)<>S
 GOTO *LP1
*WRITE
 PUT# &H1B,&H57  "ESC W
 S0=A^&HF000     "ｱﾄﾞﾚｽ(hex表記)をｱｽｷｰｺｰﾄﾞで出力
 S0=S0/&H1000
 GOSUB *WRITE1
 S0=A^&H0F00
 S0=S0/&H0100
 GOSUB *WRITE1
 S0=A^&HF0
 S0=S0/&H10
 GOSUB *WRITE1
 S0=A^&H0F
 GOSUB *WRITE1
 S0=S^&HF000     "ﾃﾞｰﾀをｱｽｷｰｺｰﾄﾞに変換して出力
 S0=S0/&H1000
 GOSUB *WRITE1
 S0=S^&H0F00
 S0=S0/&H0100
 GOSUB *WRITE1
 S0=S^&HF0
 S0=S0/&H10
 GOSUB *WRITE1
 S0=S^&H0F
 GOSUB *WRITE1
 PUT# &H0D
 RETURN
*WRITE1
 IF S0>9 GOSUB *A_F
 ELSE_GOSUB *0_9
 PUT# S0
 RETURN
*A_F
 S0=&H37+S0
 RETURN
*0_9
 S0=&H30+S0
 RETURN
 '************
*RS_RCV
 R=GET#(0)       "入力待ち
 PRINT R
 OUT R,0
 GOTO *RS_RCV
```

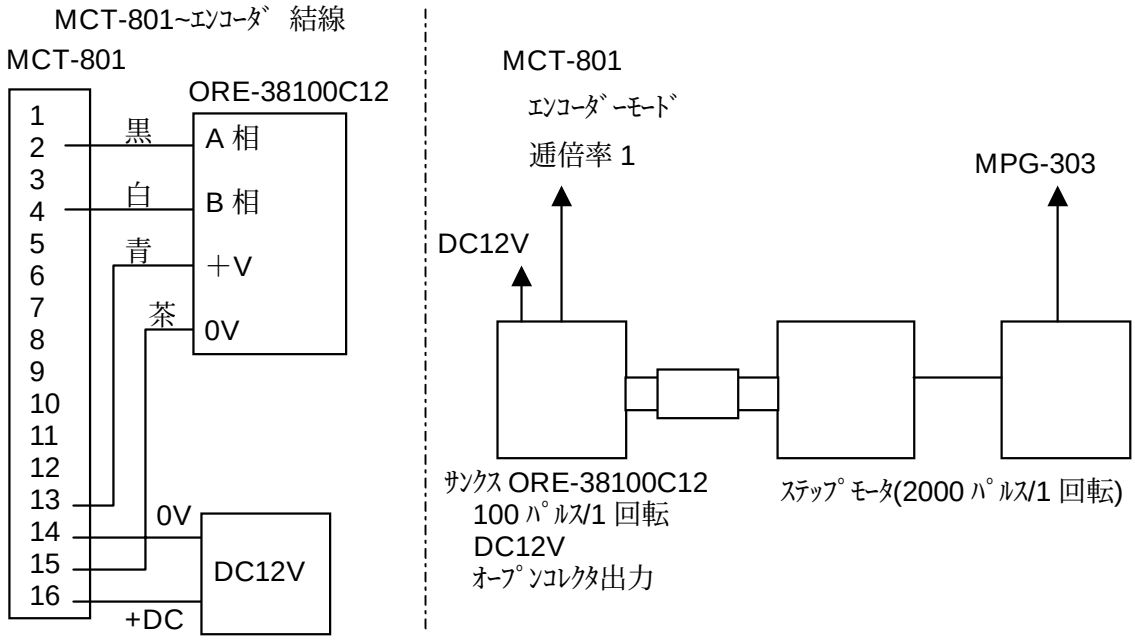
●SAMPLE 2

```
 '************
 ' GP-H70
 '  SAMPLE
 '------------
 'USE GP
 ' READ/WRITE
 '   COMMAND
 '============
 'GPH70S20
 '  980818
 '************
 S=0
 CNFG# 4,0,2
*LP1             "READ,WRITEとも同一タスク
 A=600           "書込みアドレス
 S=IN(0)         "書込みデータ
 GOSUB *WRITE
 '
 FOR A=500 TO 503  "A=読込みアドレス
 GOSUB *READ
 NEXT A
 '
 GOTO *LP1
*WRITE
 PUT# &H1B,&H57
 S0=A^&HF000
 S0=S0/&H1000
 GOSUB *WRITE1
 S0=A^&H0F00
 S0=S0/&H0100
 GOSUB *WRITE1
 S0=A^&HF0
 S0=S0/&H10
 GOSUB *WRITE1
 S0=A^&H0F
 GOSUB *WRITE1
 S0=S^&HF000
 S0=S0/&H1000
 GOSUB *WRITE1
 S0=S^&H0F00
 S0=S0/&H0100
 GOSUB *WRITE1
 S0=S^&HF0
 S0=S0/&H10
 GOSUB *WRITE1
 S0=S^&H0F
 GOSUB *WRITE1
 PUT# &H0D
 RETURN
*WRITE1
 IF S0>9 GOSUB *A_F
 ELSE_GOSUB *0_9
 PUT# S0
 RETURN
*A_F
 S0=&H37+S0
 RETURN
*0_9
 S0=&H30+S0
 RETURN
 '************
*READ
 PUT# &H1B,&H52  "ESC R
 S0=A^&HF000     "ｱﾄﾞﾚｽ(hex表記)をｽｽｷｰｺｰﾄﾞで出力
 S0=S0/&H1000
 GOSUB *READ1
 S0=A^&H0F00
 S0=S0/&H0100
 GOSUB *READ1
 S0=A^&HF0
 S0=S0/&H10
 GOSUB *READ1
 S0=A^&H0F
 GOSUB *READ1
 PUT# &H30,&H30,&H30
 PUTS# 1          "読込みデータ数＝１
 PUT# &H0D
 '
 SKIP# &H41       "Aまで読み飛ばし
 INPUT# D         "CRまで入力
 O=A-500
 OUT D,O
 RETURN
*READ1
 IF S0>9 GOSUB *A_F
 ELSE_GOSUB *0_9
 PUT# S0
 RETURN
```

08800　MPC-816

ＭＰＣ＋ＭＰＧ＋ＭＣＴでカウンタが正常でない（980902）

ＭＰＣ－８１６＋ＭＰＧ－３０３（基板ＣＥＰ－００９Ｅ）＋ＭＣＴ－８０１の組み合わせではＭＣＴ－８０１が正常に動作しないことがわかりました。ＭＰＧ－３０３のＢＵＳにＣＬＫ信号が通っていないのが原因です。ＭＩＦ－８１６との組み合わせは問題ありません。
やむおえずＭＰＧと組み合わせなければならないときは、ＭＰＧのＪ５（５番ピン）とＪ１（５番ピン）を短絡して下さい。



○プログラム（約６００ｒｐｍでモータは回転します）

```
10      PG 1                                    RUN
20      MODE 5                                   1000
30      ACCEL 20000                              2000
40      FEED 0                                   3000
50      SETPOS 0,0                               4000
60       A=C(-1)                                 5000
70      *LOOP                                    6000
80      FOR I=1 TO 10                            7000
90       RMOV -20000,0                           8000
100      PRINT C(1)                              9000
110     NEXT I                                   10000
130     MOVE 0,0                                 0
140     PRINT C(1)                               1000
150     GOTO *LOOP
```

08900　MPC-816K

816X で正常に動作していた MIO-240E の出力が 816K で挙動不審(980904)

【不具合内容】
８１６Ｘで正常に動作していたＭＩＯ－２４０Ｅの出力が８１６Ｋで安定した動作をしない。
ＭＩＯ－２４０Ｅ出力の接続先はＡ／Ｄコンバーター。

○確認バージョン

```
TNYFSC(R) Rev-2.50d [VER-Zmx1729]
 Copyright(C)by ACCEL CORP/BC-SOFT
 [255point MPC-816K MODE1~4]K8b3
```

○プログラム（抜粋）

```
OUT &H89,&H83
OUT &HFF,&H80
OUT &HFF,&H81
'
OUT &HE7,15
OUT &H00,14
OUT &H07,15
OUT &H0F,15
OUT &H07,15
'
OUT H1,14
OUT &H06,15
OUT &H0E,15
OUT &H06,15
'
OUT L1,14
OUT &H07,15
OUT &H0F,15
OUT &H07,15
```

【原因】
８１６Ｋのプログラム実行は８１６Ｘより高速で、上記のように連続ＯＵＴのＭＩＯ－２４０Ｅの出力幅は８１６ＫでＭａｘ４０μＳｅｃ（８１６ＸはＭｉｎ１００μ）程度です。
出力幅が狭すぎたため、Ａ／Ｄコンバーターが正常に動作しませんでした。
ＭＩＯの初期設定を除く各ＯＵＴコマンドの後ろにＴＩＭＥを入れて解決しました。

GET#(0)で１ﾊﾞｲﾄ以上の数値を入力する（GETN#の代用）(980908)

GET#(0)は１ﾊﾞｲﾄ入力ですからGETN#ｺﾏﾝﾄﾞの様に、一回で１ﾊﾞｲﾄを超える数値を入力することはできません。
次のﾌﾟﾛｸﾞﾗﾑは３ﾊﾞｲﾄ固定長のﾃﾞｰﾀを入力して変数に変換します。

```
10      CNFG# 4,0,2
20      *LOOP
30      TIME 10
40      PRINT RS(1)
50       R=RS(1)
60      IF R<3 THEN *LOOP
70      FOR R0=1 TO R
80        R1=GET#(0)
90        AR(R0)=R1^&H0F
100     NEXT R0
110      AR(1)=AR(1)*100
120      AR(2)=AR(2)*10
130      A=AR(1)+AR(2)
140      A=A+AR(3)
150     PRINT A
```

ＧＥＴＮ＃コマンドの不具合対応

```
TNYFSC(R) Rev-3.50j [VER-PmaX2044]
 Copyright(C)by ACCEL CORP/BC-SOFT
 [300p MPC-816K  MPG  MODE5|6]K8b5

 TNYFSC(R) Rev-2.50d [VER-Zmx1729]
 Copyright(C)by ACCEL CORP/BC-SOFT
 [255point MPC-816K MODE1~4]K8b5
```

09200　MPC-816K

プログラム実行時間(981021)

ＩＮ（）、ＳＷ（）、変数代入を含むＦＯＲ～ＮＥＸＴ１００００回ループの所要時間（①～③）から、空ループ１００００回（④）の時間を引いてそれぞれの実行時間を算出しました。
実際は、ＭＰＣのコマンドの実行効率はタスク数やその内容により変化しますから、実機で計測してみないと意味はありません。

【実験に用いたＭＰＣ】
ＭＰＣ－８１６Ｋ

```
TNYFSC(R) Rev-3.50j [VER-PmaX2044]
 Copyright(C)by ACCEL CORP/BC-SOFT
 [300p MPC-816K  MPG  MODE5|6]K8b5
```

【結果】
①ＩＮ（）関数

```
10      PUT &H15
20      FOR I=1 TO 10000
30        A=IN(0)
40      NEXT I
50      PUT &H16
950mSec
(950-320)/10000=0.063mSec
```

②ＳＷ（）関数　（８１６ＫのＳＷ（）には５ｍＳｅｃフィルタはありません）

```
10      PUT &H15
20      FOR I=1 TO 10000
30        A=SW(0)
40      NEXT I
50      PUT &H16
1070mSec
(1070-320)/10000=0.075mSec
```

③変数代入

```
10      PUT &H15
20      FOR I=1 TO 10000
30      A=50000
40      NEXT I
50      PUT &H16
730mSec
(730-320)/10000=0.041mSec
```

④空ループ

```
10      PUT &H15
20      FOR I=1 TO 10000
40      NEXT I
50      PUT &H16
320mSec
```

ポイントデータの演算とコピー(981028)

ポイントデータの演算（２つの座標値を合成してその座標にＭＯＶＥする）

```
>HOME 0                    ←現在点を0ｸﾘｱ
>X(1)=100                  ←座標値ｾｯﾄ
>PRINT X(1)                ←確認表示
 100
>X(2)=200                  ←座標値ｾｯﾄ
>PRINT X(2)                ←確認表示
 200
>A=X(1)+X(2)               ←X(1)とX(2)を足してAに代入
>PRINT A                   ←確認表示
 300
>MOVE A,0                  ←X=300,Y=0のﾎﾟｲﾝﾄへ移動
>PRINT X(0)                ←現在点の確認表示
 300
```

ポイントデータのコピー（あるポイントデータを別のポイントデータにコピーする）

```
>SETP 1 123 456            ←X(1)=123,Y(1)=456をｾｯﾄ
>PRINT P(1)                ←P(1)の確認表示
 123 456 0 0
>PRINT P(100)              ←P(100)の確認表示
 0 0 0 0
>SETP 100,X(1),Y(1)        ←X(1)をX(100)、Y(1)をY(100)にｺﾋﾟｰ
>PRINT P(100)              ←P(100)の確認表示
 123 456 0 0
>
```

09400　MPC-816

Ａ／Ｄ、Ｄ／Ａコンバーターについて(981216)

現在のところ弊社純正のＡ／Ｄ、Ｄ／Ａコンバーターはありませんが、日本コンピューター・ネットワーク株式会社（関西地区）、ソフマート株式会社（関東地区）で販売している「ＧＵＰＰＹ　ＳＥＲＩＳＥ」で使用できるボードがあります。
弊社で動作確認をし、すでに実績のあるコンバーターとして次のものがあります。

「ＧＰＹ－１４」　８ビット８ＣＨ　Ａ／ＤコンバーターＣＡＲＤ
「ＧＰＹ－１５」　８ビット２ＣＨ　Ｄ／ＡコンバーターＣＡＲＤ

＊＊＊＊＊

■Ａ／Ｄコンバーター

「ＧＰＹ－１４」の使用例です。ＧＰＹ－１４のアドレスはデフォルトで&H40～&H43になっていますが、これはＭＩＦ－８１６のパルスポートと重なります。使用する際は変更してください。

●サンプルプログラム

３チャンネルの電圧を読み込みます。

接続

ｱﾅﾛｸﾞ入力ｺﾈｸﾀ	
PIN	信号
7	VIN3
8	A-GND

＋ → 電源(DC0～5V)
GND

```
*LOOP
 OUT 255,0
 GOSUB *GPY
 OUT 0,0
 GOSUB *GPY
 GOTO *LOOP
*GPY
 OUT 0,&H43        ←A/D動作開始
 TIME 1
 A=IN(&H43)        ←ﾃﾞｰﾀ読込み
 PRINT A
 RETURN
```

■Ｄ／Ａコンバーター

「ＧＰＹ－１５」の使用例です。ＧＰＹ－１５のアドレスはデフォルトで&H50～&H53になっていますが、これはＭＩＦ－８１６のＩ／Ｏポートと重なります。使用する際は変更してください。
下記のプログラムはアドレス設定ジャンパーＪ０を「３」（＆Ｈ５８～＆Ｈ５Ｂ）に変更して作成しました。

●出力例

【ﾊﾟｲﾎﾟｰﾗ出力（初期設定）】

```
OUT 0,&H58 でVOUT0 ～ A-GND間 0V
OUT &H80,&H58 でVOUT0 ～ A-GND間 +5V
OUT &H7F,&H58 でVOUT0 ～ A-GND間 -4.96V
```

【ﾕﾆﾎﾟｰﾗ出力（ｼﾞｬﾝﾊﾟ-J1,J2,J3,J4をB側）】

```
OUT 0 &H58        でVOUT0 ～ A-GND間 0V
OUT &HFF &H58     でVOUT0 ～ A-GND間 -5V
```

●サンプルプログラム

接続

出力ｺﾈｸﾀ	
PIN	信号
1	+12V
7	A-GND
9	-12V
19	A-GND
21	VOUT0

+12 → 電源(±12V)
COM → 電源(±12V)
-12 → 電源(±12V)
A-GND → Ⓥ 電圧計

【ﾊﾟｲﾎﾟｰﾗ出力】

ＶＯＵＴ０の電圧が０Ｖから５Ｖへ上昇します。

```
OUT &H80,&H5B     ←A,B,Cﾎﾟｰﾄ初期設定
FOR I=&H7F TO 0
  I1=I+&H80
 OUT I1,&H58      ←Aﾎﾟｰﾄに出力
 TIME 10
DEC I
```

【ﾕﾆﾎﾟｰﾗ出力】

ＶＯＵＴ０の電圧が０Ｖから－５Ｖへ変化します。

```
OUT &H80,&H5B
FOR I=&H00 TO &HFF
 OUT I,&H58
TIME 10
NEXT I
```

■補記
①これらのボードをＭＰＣラックに挿入するにはアダプター「ＡＤＰ－３２５」が必要です。
②この他に 「ＧＰＹ－６７」 ＬＣＤモジュールインターフェースカード の動作も確認してあります。
③弊社ではＧＰＹシリーズは取り扱っておりません。販売元、取り扱い商社へお問い合わせ下さい。
販売元（トランジスタ技術１９９８／１２月号より）
日本コンピューター・ネットワーク株式会社 〒660-0083 兵庫県尼崎市道意町7-1-3 TEL06-415-2638
ソフトマート株式会社 〒101-0063 東京都千代田区神田淡路町2-4 ﾕﾆｵﾝﾋﾞﾙ8F TEL03-3256-5881

09500　MPC-816

ＹＰＬＳのスローアップ、スローダウンについて(981224)

ＹＰＬＳは第２パラメータの値を変更すると周波数が変わります。これを利用して擬似的なスローアップ、スローダウンができます。
サブルーチン ＊Ｓ＿ＵＰ は 変数Ｂの値を加算しパルス周波数を上げています。＊Ｓ＿ＤＯＷＮはその逆です。
変数Ｂの変化量や間隔、最高最低スピードはカット＆トライです。
この方法で見た目は台形駆動の様にみえますが、実用的かどうかはイマイチ？？？
また、この方法では加減速区間のパルス出力数は不定です。

```
10      *LOOP
20      PLS_MIF
30       M0=100        ←最低周波数
40       M1=4000       ←最高周波数
50       A=1
60       B=M0
70       C=0
80      YPLS A,B,C
90      GOSUB *S_UP
100     WAIT SW(0)=1
110     GOSUB *S_DOWN
120      A=0
130     WAIT A=256
140     WAIT SW(0)=0
150     GOTO *LOOP
160     END
170     *S_UP
180      B=B+10
190     PRINT B     ←ﾃﾞﾊﾞｯｸﾞ表示。ﾃﾞﾊﾞｯｸﾞ後削除
200     IF B<M1 THEN *S_UP
210     RETURN
220     *S_DOWN
230      B=B-10
240     PRINT B     ←ﾃﾞﾊﾞｯｸﾞ表示。ﾃﾞﾊﾞｯｸﾞ後削除
250     IF B>M0 THEN *S_DOWN
260     RETURN
```

※190 と 240 の PRINT 文があるとｽﾑｰｽﾞに動作しません。本番では削除してください。

09600　MPC-816K

ＭＰＣ－８１６Ｋのプログラム実行スピード(990118)

複数のタスクを起動した状態で、ＳＷ（）・ＩＦ・ＯＮ・ＯＦＦを含むループを１００００回実行してタイムを計りました。

【測定したＭＰＣ－８１６Ｋ】

```
TNYFSC(R) Rev-3.50p[VER-PmaX2044]
Copyright(C)by ACCEL CORP/BC-SOFT
[300p MPC-816K  MPG  MODE5|6]K8b5
```

【実験プログラム】

```
10      FOR I=1 TO 5
20       FORK I,*DUMMY
30      NEXT I
40       I=0
50      PUT &H15    ←ｽﾀｰﾄ
60      *LP
70       I=I+1
80      IF I>10000 THEN *END
90       A=SW(0)
100     IF A<>0 THEN *ONLED
105        ELSE_THEN *OFFLED
110     *ONLED
115     ON 0
117     GOTO *LP
120     *OFFLED
125     OFF 0
130     GOTO *LP
140     *END
150     PUT &H16    ←ｽﾄｯﾌﾟ
160     END
170     *DUMMY      ←ﾀﾞﾐｰﾀｽｸ
180     ON 1
190     TIME 10
200     OFF 1
210     TIME 10
220     IF SW(1)=1 THEN *DUMMY
230      B=IN(0)
240     IF B<>0 THEN *DUMMY
250     GOTO *DUMMY
```

【測定結果】

約 13000mSec

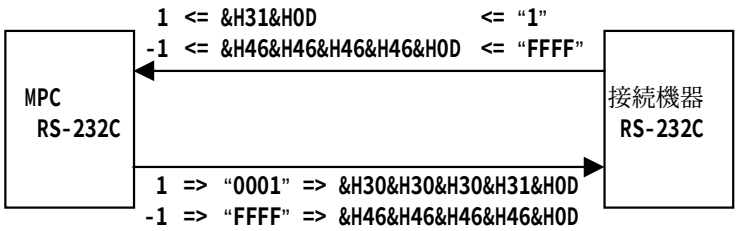
ﾙｰﾌﾟ１回では 1.3mSEC

【補記】

ＭＰＣ－８１６ＫのＳＷ（）文には５ｍＳｅｃフィルタは入っていません。

09700　MPC-816K

アスキーコード文字列を２バイト１０進数に変換(990310)




```
*__INIT
        CNFG# 4,0,2
*MAIN
        GOSUB *GET
        GOSUB *PUT
        GOTO *MAIN
        '============
*GET                                    "入力
        FOR I0=0 TO 31
                AR(I0)=0
        NEXT I0
        R1=-1
*GET_01
        R0=GET#(0)                      "入力は可変長
        IF R0=&HD THEN *GET_END         "ﾀﾐﾈｰﾀ=&HD で終り
        IF R0<&H3A GOSUB *GET_09        "0~9 なら
                ELSE_GOSUB *GET_AF      "A~F なら
        R1=R1+1
        AR(R1)=R0                       "1ﾊﾞｲﾄづつ AR()に格納
        GOTO *GET_01
*GET_END
        R2=1
        FOR I0=R1 TO 0
                AR(I0)=AR(I0)*R2        "桁ｼﾌﾄ
                R2=R2*&H10
        DEC I0
        R3=0
        FOR I0=0 TO R1
                R3=R3+AR(I0)            "合体
        NEXT I0
        R4=R3^&H8000                    "ﾄｯﾌﾟﾋﾞｯﾄﾁｪｯｸ
        IF R4<>0 GOSUB *GET_CONV
        'GET=
        PRINT STR(-1),R3                "入力
        GOTO *__RTN
*GET_09
        R0=R0-&H30
        GOTO *__RTN
*GET_AF
        R0=R0-&H37
        GOTO *__RTN
*GET_CONV
        R3=R3-65536
        GOTO *__RTN
        END
```

```
        '============
*PUT                                    "出力
        IF R3<0 GOSUB *PUT_CONV
        R1=&H1000
        R2=&H0FFF
        FOR I0=0 TO 3                   "出力は 2 ﾊﾞｲﾄ固定長
                AR(I0)=R3/R1            "桁ｼﾌﾄ
                R3=R3^R2
                R1=R1/&H10
                R2=R2/&H10
        NEXT I0
        'PUT=
        PRINT STR(-1)
        FOR I0=0 TO 3
                IF AR(I0)<10 GOSUB *PUT_09
                        ELSE_GOSUB *PUT_AF
                PUT# R3                 "出力
                PUT R3
        NEXT I0
        PUT# &HD
        PUT &HD
        GOTO *__RTN
*PUT_CONV
        R3=R3-65536
        R3=R3^65535
        GOTO *__RTN
*PUT_09
        R3=AR(I0)+&H30
        GOTO *__RTN
*PUT_AF
        R3=AR(I0)+&H37
        GOTO *__RTN
        '============
*__RTN
        RETURN
```